MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2019 / 2020 SESSION

DPJ 5018 – PROGRAMMING IN JAVA

(DIT Only)

9 MARCH 2020
2.30 p.m. – 4.30 p.m.
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This question paper consists of 12 pages.
2. There are **TWO (2)** sections:
   - **Section A:** Three (3) questions. Each question carries 10 marks. Answer ALL questions.
   - **Section B:** Two (2) questions. Each question carries 20 marks. **Choose ONE (1) question only.**
3. Total mark is **50** marks.
4. Please write your answers in the **Answer Booklet** provided.

SECTION A: STRUCTURED QUESTIONS (30 MARKS)

Answer ALL questions. Each question carries 10 marks.

QUESTION 1 [10 MARKS]

Refer to the following two-dimensional array declaration.

```
int [][] poll = {{113,107,124}, {105,139,126}};
```

The array represents the numbers of polls conducted randomly from different social medias to know who are with Team Jedi or Team Sith.

Team/Media	Facebook	Instagram	Twitter
Jedi	113	107	124
Sith	105	139	126

Table 1: Numbers of poll

Write a method named `getMinimumPoll()` that accepts the array as the argument and displays the minimum poll of team and from which media. This method is not returning any value. The following declarations are done for you:

```
String []team = {"Jedi","Sith"};
String []media = {"Facebook","Instagram","Twitter"};
```

You may refer the following output: (10 Marks)

```
run:
Team Sith got the minimum poll of 105 from Facebook
```

[TOTAL 10 MARKS]

Continued...

QUESTION 2 [10 MARKS]

a) Refer to the following class declaration and answer the following questions. (5 Marks)

```
public class Heroes
{
    private String heroName, teamName;

    public Heroes()
    {}

    public String getHeroName()
    {
        return heroName;
    }
   //some other code here
}
```

i. Write a mutator method for the `heroName` field. *[2.5m]*

ii. Write a `toString()` method that returns the descriptions of hero name and team name. *[2.5m]*

b) What is the output of ***Program 1*** (***MainGhost.java***)? (5 Marks)

```
class Ghost
{   public Ghost()
    {
        System.out.println("Ghost movie.");
    }
    public Ghost(String name)
    {
        this();
        System.out.println("The ghost name is "+name);
    }
}
class Casper extends Ghost
{
    public Casper(String text)
    {
        System.out.println(text);
    }
}
```

Continued...

```
class HantuKakLimah extends Ghost
{
    public HantuKakLimah()
    {
        System.out.println("Kak Limah is in KL.");
    }
    public HantuKakLimah(String text)
    {
        super(text);
        System.out.println("She has returned home.");
    }
}
public class MainGhost
{
    public static void main(String[] args)
    {
        Casper obj1 = new Casper("A friendly ghost.");
        HantuKakLimah obj2 = new HantuKakLimah("Kak
Limah.");

     }
}
```

Program 1: MainGhost.java

[TOTAL 10 MARKS]

Continued...

QUESTION 3 [10 MARKS]

a) Refer to ***Program 2 (TrialTest.java)*** for the following questions.

```
import java.util.Scanner;
public class TrialTest
{
    public static void main (String [] arg)
    {
        String passcode ="";
        boolean loop=true;

     do {
         try {
         System.out.print("Set your passcode (5 digits): ");
         passcode = new Scanner(System.in).next();

         if(passcode.length()<5 || passcode.length()>5)
            throw new IllegalArgumentException("Access
Denied!");

         else {  System.out.println(passcode+": Passed!");
                 loop = false;
         }

         int value = Integer.parseInt(passcode);
         System.out.println("The number is "+value);

         }

         catch(IllegalArgumentException e) {
             System.out.println(passcode+":
"+e.getMessage());
             System.out.println("Try again");
         }

         catch(RuntimeException e) {
             System.out.println("Task could not perform.");
             System.out.println("Try again");
             loop = true;
         }

     }while(loop);

     System.out.println("Thank you!");
   }
}
```

Program 2: TrialTest.java

Continued...

Write the output of the program if user enters: (5 Marks)

i. ***87241*** for `passcode`. *[2.25m]*

ii. ***719*** for `passcode` (first loop), then ***719ab*** (second loop) and lastly ***71943*** (third loop). *[2.75m]*

b) Write the code for the following questions: (5 Marks)

i. Create a text field with font specifications: style – italic, size -15, font name – Arial. *[1.5m]*

ii. Create a checkbox button with the text 'Agree' and it initially appears selected. *[1m]*

iii. Create a button with an icon (file name: edit.jpg). Set the button's background colour to yellow. *[2.5m]*

[TOTAL 10 MARKS]

Continued...

SECTION B: APPLICATION QUESTION (20 MARKS)

Choose ONE question only.

QUESTION 1 [20 MARKS]

In a college, assessment for a subject is consisting of coursework (50%) and final exam (50%). The coursework's components are:

- Midterm test (15%)
- Project (15%)
- Quiz (10%)
- Lab (10%)

You are required to write a program to demonstrate this situation. Refer to the following UML class diagrams as shown in ***Figure Q1a*** and the following descriptions.

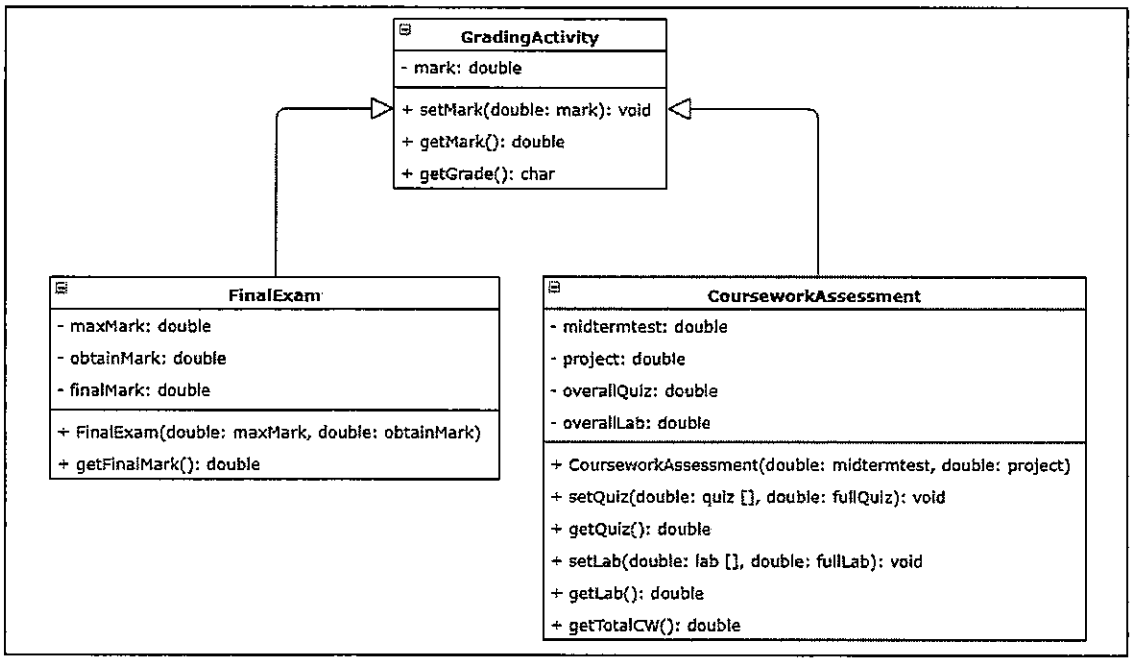

Figure Q1a: UML class diagrams for `GradingActivity`, `FinalExam` *and* `CourseworkAssessment`

Continued...

a) Write a class named `GradingActivity`. The class consists of the following data fields and methods.

- The `mark` data field represents the overall marks that a student achieved for a subject.
- Method `setMark` accepts a mark as the parameter and sets it accordingly.
- Method `getMark` returns the overall mark of the subject.
- Method `getGrade` determines the grade for this subject based on the mark and returns the grade. Refer to the following table (***Table 1***):

Mark	Grade
85.0 and above	A
70.0 – less than 85.0	B
60.0 – less than 70.0	C
50.0 - less than 60.0	D
40.0 - less than 50.0	E
Less than 40.0	F

Table 1: Grading scheme

b) Write a class named `CourseworkAssessment` that inherits `GradingActivity`. The class consists of the following data fields, constructor and methods.

Class/Instance fields:

- The `midtermtest`, `project`, `overallQuiz` and `overallLab` field represents the mark of each components in coursework assessment that a student achieved for a subject.

Constructor:

- Constructor `CourseworkAssessment` constructs an object with parameters: `midtermtest` and `project`, and set them according to the class data fields.

Methods:

- Method `setQuiz` calculates the total quiz marks and the mark obtained for the 10% assessment. This method accepts a `double` array of quizzes mark and the full mark of all quizzes.
- Method `getQuiz` returns the overall quiz mark.
- Method `setLab` calculates the total lab marks and the mark obtained for the 10% assessment. This method accepts a `double` array of lab exercises marks and the full mark of all labs.

Continued...

- Method `getLab` returns the overall lab mark.
- Method `getTotalCW` calculates the total coursework mark that adding all the marks of midterm test, project, overall quiz mark and overall lab marks.

c) Write a class named `FinalExam` that inherits `GradingActivity`. The class consists of the following data fields, constructor and methods:

Class/Instance fields:

- The `maxMark` field represents the full mark of the exam paper.
- The `obtainMark` field represents the mark that the student obtained for the exam paper.
- The `finalMark` field represents the result mark that the student obtained for the assessment.

Constructor:

- Constructor FinalExam constructs an object with parameters: `maxMark` and `obtainMark`, and set them according to the class data fields.

Methods:

- Method `getFinalMark` calculates the result mark for the 50% assessment and returns the mark. ***Formula:*** mark that the student obtained divided with the full mark of the paper and multiply by 50.

d) Write an application program to demonstrate these classes.

- Prompt the user to key-in:
  - Student name
  - Student ID
  - Subject code and name
  - Midterm test mark
  - Project mark
- Next, ask the user to key in number of quizzes for this subject and the full mark (total) of all the quizzes.
- Then, create an array to store the quizzes mark based on the number of quizzes.
- Ask user to key-in the mark for each quiz and store it in the array.
- Next, same goes for the lab. Ask the user to key in number of lab exercises for this subject and the full mark (total) of all the labs.
- Create an array to store the labs mark based on the number of lab exercises.

Continued...

- Ask user to key-in the mark for each lab and store it in the array.

- Create a `CourseworkAssessment` object with the parameters of the midterm test mark and the project mark.
- Call method `setQuiz` and pass the array and the full mark of the quizzes.
- Call method `setLab` and pass the array and the full mark of the lab exercises.
- Display the quiz mark (of 10%) and the lab mark (of 10%) by calling the appropriate methods.

- Next, prompt the user to key-in the final exam mark details: the full mark of the paper and the mark obtained by the student.
- Create a `FinalExam` object with the parameters: full mark of the paper and the obtained.
- Display the student and the subject info.
- Display the total coursework mark (50%) by calling the appropriate method.
- Display the total final exam mark (50%) by calling the appropriate method.
- Call method `setMark` and pass the overall mark of the subject (total coursework add total final mark).
- Display the overall mark and the grade calling the appropriate methods.

Refer to ***Figure Q1b*** for the sample output screen.

Continued...

```
Enter student name: Yuna Ko
Enter student ID: s119203865
Subject code & name: DPJ5018 Java

Coursework Details: (Enter the marks)
Midterm test mark (15%): 13.75
Project Mark (15%): 14
How many quizzes? 3
Full mark (total) of all quizzes? 45
Quiz 1: 10.5
Quiz 2: 15
Quiz 3: 7.5
How many lab exercise? 5
Full mark(total) of all lab exercises? 50
Lab 1: 10
Lab 2: 10
Lab 3: 8
Lab 4: 9
Lab 5: 8.5

Overall quiz mark (10%) = 7.33
Overall lab mark (10%) = 9.10

Final Exam Details:(Enter the marks)
Full mark of the exam paper: 50
Mark obtained: 41.25

Final Result
-----------------
Student ID = s119203865
Student name = Yuna Ko
Subject code & name = DPJ5018 Java
Total CW (50%) = 44.18
Final exam (50%)= 41.25
Total Overall Mark (100%)= 85.43
Grade = A
```

Entered by user

Figure Q1b: Output screen

[TOTAL 20 MARKS]

Continued...

QUESTION 2 [20 MARKS]

Write a GUI application that translates a greeting (in English) to a selected language chosen by user. User chooses the greeting form a drop-down menu, then selects the language and clicks the button.

The default text is "Greeting", after user chooses the greeting, the text is changed to the greeting accordingly. Additionally, once user clicks the button, the greeting is translated to the selected language, which is displayed using another text.

Refer to ***Figure Q2a*** for details layout.
Refer to ***Figure Q2b***, ***Q2c***, and ***Q2d*** for the sample output screens.

Other descriptions are as follows:

- Greeting messages:
  - *Hello*
  - *Goodbye*
  - *Thank you*

- Translated greetings:

Language / Greeting	Japanese	Korean	Arabic
Hello	Konnichiwa	Annyeonghaseyo	Ahlan
Goodbye	Sayonara	Annyeonghi gyeseyo	Ma'as-salama
Thank you	Arigato	Kamsahamnida	Syukran

- Image file name:
  - Japanese: `geisha.png`
  - Korean: `korea.png`
  - Arabic: `arab.png`

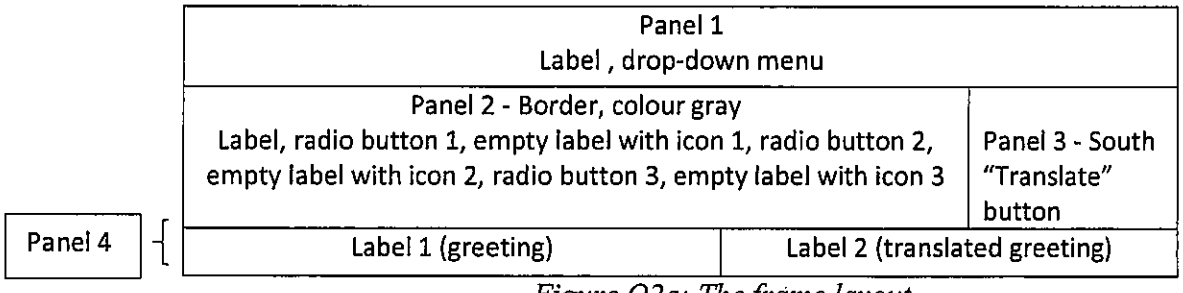
Panel 1
Label , drop-down menu
Panel 2 - Border, colour gray
Label, radio button 1, empty label with icon 1, radio button 2, empty label with icon 2, radio button 3, empty label with icon 3
Panel 3 - South "Translate" button
Panel 4
Label 1 (greeting)
Label 2 (translated greeting)

Figure Q2a: The frame layout

Continued...

- The frame layout (Figure Q2a):
  - Panel 1 is located at the *NORTH* of the frame.
  - Panel 2 is located at the *CENTER* of the frame.
  - Panel 3 with button at *BorderLayout.SOUTH* is located at the *EAST* of the frame.
  - Panel 4 is located at the *SOUTH* of the frame.

- Font's labels at Panel 4:
  - Lucida Calligraphy, bold and italic, size 25.
  - Label 1 colour is blue.
  - Label 2 colour is orange.

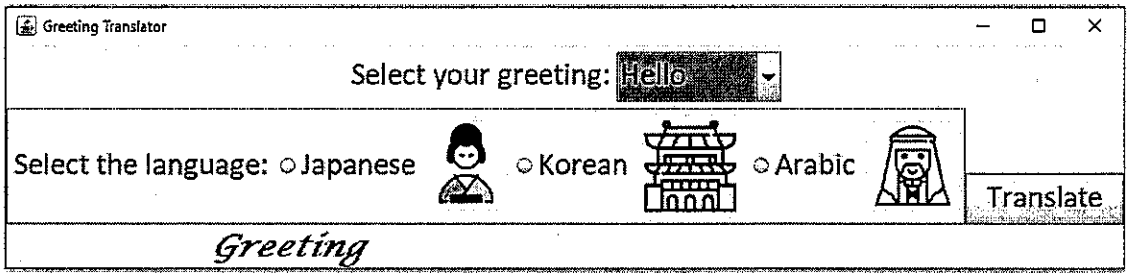

Figure Q2b: Sample output 1

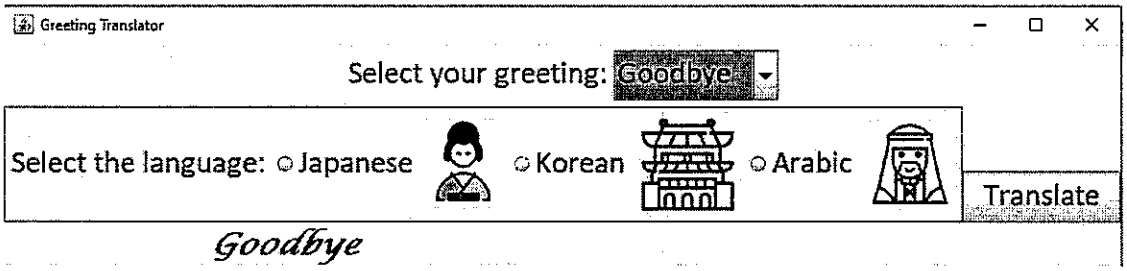

Figure Q2c: Sample output 2 (After user chooses the greeting)

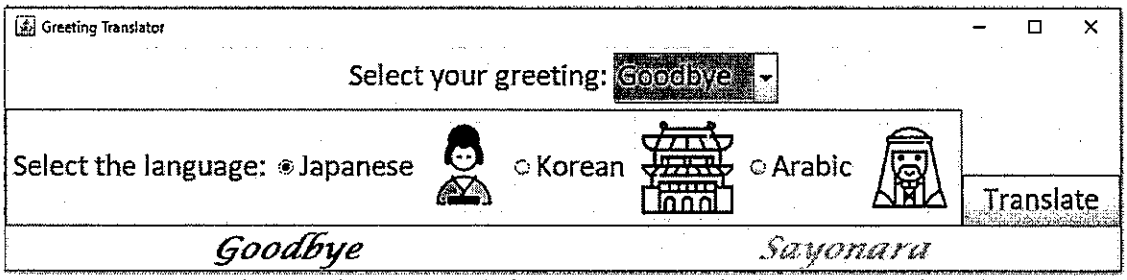

Figure Q2d: Sample output 3 (After user selects the language and clicks the button)

[TOTAL 20 MARKS]

End of Page.